

REMARKS

In accordance with the foregoing, claims 16-24 have been amended. No new matter is believed to be presented, and approval and entry are respectfully requested.

Claims 16-24 are pending and under consideration. Reconsideration of the claims is respectfully requested.

Rejection Under 35 U.S.C. §102(a)

In the Office Action at page 2, claims 16-24 were rejected under 35 U.S.C. §102(a) as being unpatentable over U.S. Patent No. 5,991,972 to Nageswaran in view of newly cited U.S. Patent No. 6,487,578 to Ranganathan. This rejection is respectfully traversed and reconsideration is requested.

Claims 16-18

Independent claim 16 is directed to a method for controlling a plurality of threads that perform parallel processing. In relevant part, amended independent claim 16 recites "counting a number of running threads performing parallel processing using a running thread counter and a number of standby threads that are in a standby state using a standby thread counter for a predetermined time period", "setting a maximum number of running threads in accordance with the number of running threads during the predetermined time period using a maximum running thread counter", "incrementing the number of running threads using the running thread counter and decrementing the number of standby threads using the standby thread counter in response to a run request", and "updating the maximum number of running threads set by the maximum running thread counter if the incremented number of running threads is greater than the maximum number of running threads".

Independent claim 17 is also directed to a method for controlling a plurality of threads that perform parallel processing. Independent claim 17 has been amended in a manner similar to independent claim 16 and recites, in relevant part, "counting a number of running threads performing parallel processing using a running thread counter and a number of standby threads that are in a standby state using a standby thread counter for a predetermined time period", "setting an average number of running threads in accordance with the number of running threads during the predetermined time period using an average running thread counter", "incrementing the number of running threads using the running thread counter and decrementing the number of standby threads using the standby thread counter in response to a run request", and "updating the average number of running threads set by the average running thread counter if the incremented number of running threads is greater than the average number

of running threads.

Independent claim 18 is also directed to a method for controlling a plurality of threads that perform parallel processing. Independent claim 18 has been amended in a manner similar to independent claims 16 and 17 and recites, in relevant part, "counting a number of running threads performing parallel processing using a running thread counter and a number of standby threads that are in a standby state using a standby thread counter for a predetermined time period", "setting a product obtained by multiplying the number of running threads during the predetermined time period by a predetermined coefficient using a product running thread counter", "incrementing the number of running threads using the running thread counter and decrementing the number of standby threads using the standby thread counter in response to a run request", and "updating the product set by the product running thread counter if the incremented number of running threads is greater than the product".

At page 2, numbered item 4, the Office Action contends that Nageswaran teaches "monitoring a number of running threads performing parallel processing and a number of standby threads that are in a standby state for a predetermined time period", "setting a necessary number of standby threads in accordance with the number of running threads during the predetermined time period", "comparing the number of the standby threads with the necessary number of the standby threads at a predetermined time interval", and "terminating a number of standby threads exceeding the necessary number when the number of standby threads is greater than the necessary number". Applicants respectfully disagree and assert that Nageswaran teaches only reducing the number of threads of the thread pool 138 to a minimum number of threads.

At page 3, numbered item 5, the Office Action contends that newly cited Ranganathan teaches that "the necessary number may be derived from statistical information pertaining to system workload, including but not limited to a maximum number of running threads; an average number of running threads; and a product obtained by multiplying the number of running threads by a predetermined coefficient". Applicants respectfully disagree. Ranganathan is directed to dynamic feedback costing to enable adaptive control of resource utilization. However, Ranganathan teaches only the modification of operating parameters based on resource utilization. Ranganathan does not teach or suggest that "the necessary number may be derived from statistical information pertaining to system workload", as asserted in the Office Action.

Further, Applicants respectfully submit that neither Nageswaran nor newly cited Ranganathan teaches or suggests "counting a number of running threads performing parallel processing using a running thread counter and a number of standby threads that are in a

standby state using a standby thread counter for a predetermined time period", "setting a maximum number of running threads in accordance with the number of running threads during the predetermined time period using a maximum running thread counter", "incrementing the number of running threads using the running thread counter and decrementing the number of standby threads using the standby thread counter in response to a run request", or "updating the maximum number of running threads set by the maximum running thread counter if the incremented number of running threads is greater than the maximum number of running threads" as recited in amended independent claim 16. Independent claims 17 and 18 recite similar features. Thus, Applicants respectfully submit that neither Nageswaran nor newly cited Ranganathan teaches or suggests setting a maximum value, an average value, or a product according to the number of running threads, and updating the maximum value, the average value, or the product value if the incremented number of running threads of a running thread counter is greater than the set maximum value, average value, or product value, as set forth in independent claims 16, 17, and 18, respectively.

For at least these reasons, Applicants respectfully submit that newly cited Ranganathan fails to cure the deficiencies of Nageswaran. Accordingly, Applicants submit that Nageswaran and Ranganathan, whether taken alone or in combination, fail to teach or suggest all of the features of amended independent claims 16, 17, and 18. Thus, Applicants submit that amended independent claims 16, 17, and 18 patentably distinguish over the prior art and are in condition for allowance.

Claims 19-21

Independent claim 19 is directed to a controller for controlling a plurality of threads that perform parallel processing. In relevant part, amended independent claim 19 recites that the thread management table includes "a running thread counter to count the number of running threads", "a standby thread counter to count the number of standby threads", and "a maximum running thread counter". Additionally, amended independent claim 19 recites that the controller includes "a comparison circuit setting a maximum number of running threads during a predetermined period in accordance with the number of running threads included in the thread information to the maximum running thread counter, and comparing the number of standby threads with the maximum number of running threads" and "a thread to cause the running thread counter to increment the number of running threads and the standby thread counter to decrement the number of standby threads in response to a run request, wherein the thread updates the maximum number of running threads set by the maximum running thread counter if the incremented number of running threads is greater than the maximum number of running

threads".

Independent claim 20 is also directed to a controller for controlling a plurality of threads that perform parallel processing. Independent claim 20 has been amended in a manner similar to independent claim 19 and recites, in relevant part, that the thread management table includes "a running thread counter to count the number of running threads", "a standby thread counter to count the number of standby threads", and "an average running thread counter". Additionally, amended independent claim 20 recites that the controller includes "a comparison circuit setting a maximum number of running threads during a predetermined period in accordance with the number of running threads included in the thread information to the average running thread counter, and comparing the number of standby threads with the average number of running threads" and "a thread to cause the running thread counter to increment the number of running threads and the standby thread counter to decrement the number of standby threads in response to a run request, wherein the thread updates the average number of running threads set by the average running thread counter if the incremented number of running threads is greater than the average number of running threads".

Independent claim 21 is also directed to a controller for controlling a plurality of threads that perform parallel processing. Independent claim 21 has been amended in a manner similar to independent claims 19 and 20 and recites, in relevant part, that the thread management table includes "a running thread counter to count the number of running threads", "a standby thread counter to count the number of standby threads", and "a product running thread counter". Additionally, amended independent claim 21 recites that the controller includes "a comparison circuit setting a product to the product running thread counter, the product being obtained by multiplying the number of running threads during a predetermined time period by a predetermined coefficient in accordance with the number of running threads included in the thread information, and comparing the number of standby threads with the product" and "a thread to cause the running thread counter to increment the number of running threads and the standby thread counter to decrement the number of standby threads in response to a run request, wherein the thread updates the product set by the product running thread counter if the incremented number of running threads is greater than the product".

At pages 3-4, numbered item 7, the Office Action contends that Nageswaran teaches a controller as claimed including "a thread management table storing thread information of the plurality of threads, wherein the thread information includes a number of running threads performing parallel processing and a number of standby threads that are in a standby state," "a thread management circuit requesting thread generation based on the number of standby

threads stored in the thread management table, and requesting a standby thread to run", "a comparison circuit setting a necessary number of standby threads in accordance with the number of running threads included in the thread information, and comparing the number of standby threads with the necessary number", and "a termination circuit terminating a number of standby threads exceeding the necessary number when the number of standby threads is greater than the necessary number."

Applicants respectfully submit, however, that Nageswaran fails to teach or suggest that the thread management table includes "a running thread counter to count the number of running threads", "a standby thread counter to count the number of standby threads", and "a maximum running thread counter" as recited in amended independent claim 19. Further, Nageswaran fails to teach or suggest "a comparison circuit setting a maximum number of running threads during a predetermined period in accordance with the number of running threads included in the thread information to the maximum running thread counter, and comparing the number of standby threads with the maximum number of running threads" or "a thread to cause the running thread counter to increment the number of running threads and the standby thread counter to decrement the number of standby threads in response to a run request, wherein the thread updates the maximum number of running threads set by the maximum running thread counter if the incremented number of running threads is greater than the maximum number of running threads", as recited in amended independent claim 19. Similar features are recited in amended independent claims 20 and 21.

At page 4, numbered item 8, the Office Action again contends that newly cited Ranganathan teaches that "the necessary number may be derived from statistical information pertaining to system workload, including but not limited to a maximum number of running threads; an average number of running threads; and a product obtained by multiplying the number of running threads by a predetermined coefficient". Applicants respectfully disagree. Ranganathan is directed to dynamic feedback costing to enable adaptive control of resource utilization. However, Ranganathan teaches only the modification of operating parameters based on resource utilization. Ranganathan does not teach or suggest that "the necessary number may be derived from statistical information pertaining to system workload", as asserted in the Office Action.

Applicants respectfully submit that neither Nageswaran nor newly cited Ranganathan teaches or suggests setting a maximum value, an average value, or a product according to the number of running threads, and updating the maximum value, the average value, or the product value if the incremented number of running threads of a running thread counter is greater than

the set maximum value, average value, or product value, as set forth in independent claims 19, 20, and 21, respectively.

For at least these reasons, Applicants respectfully submit that newly cited Ranganathan fails to cure the deficiencies of Nageswaran. Accordingly, Applicants submit that Nageswaran and Ranganathan, whether taken alone or in combination, fail to teach or suggest all of the features of amended independent claims 19, 20, and 21. Thus, Applicants submit that amended independent claims 19, 20, and 21 patentably distinguish over the prior art and are in condition for allowance.

Claims 22-24

Amended independent claims 22, 23, and 24 are directed to computer readable storage media storing programs for performing the methods of amended independent claims 16, 17, and 18, respectively. Accordingly, Applicants respectfully submit that amended independent claims 22, 23, and 24 patentably distinguish over the prior art for at least the same reasons set forth above with respect to amended independent claims 16, 17, and 18 and, therefore, claims 22, 23, and 24 are in condition for allowance.

Conclusion

In accordance with the foregoing, it is respectfully submitted that all outstanding objections and rejections have been overcome and/or rendered moot. And further, that all pending claims patentably distinguish over the prior art. Thus, there being no further outstanding objections or rejections, the application is submitted as being in condition for allowance which action is earnestly solicited.

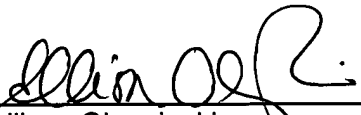
If the Examiner has any remaining issues to be addressed, it is believed that prosecution can be expedited by the Examiner contacting the undersigned attorney for a telephone interview to discuss resolution of such issues.

If there are any underpayments or overpayments of fees associated with the filing of this Amendment, please charge and/or credit the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 31 March 2006

By: 
Allison Olenginski
Registration No. 55,509

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501